Preprint
UCRL-JC-151933

Loops in Reeb Graphs of 2-Manifolds

K. Cole-McLaughlin, H. Edelsbrunner, J. Harer, V. Natarajan, V. Pascucci

This article was submitted to *ACM Symposium on Computational Geometry, San Diego, CA*
06/08/2003 – 06/10/2003

U.S. Department of Energy

Lawrence
Livermore
National
Laboratory

February 11, 2003

Loops in Reeb Graphs of 2-Manifolds *

Kree Cole-McLaughlin†, Herbert Edelsbrunner‡, John Harer§, Vijay Natarajan¶ and Valerio Pascucci†

Abstract

Given a Morse function f over a 2-manifold with or without boundary, the Reeb graph is obtained by contracting the connected components of the level sets to points. We prove tight upper and lower bounds on the number of loops in the Reeb graph that depend on the genus, the number of boundary components, and whether or not the 2-manifold is orientable. We also give an algorithm that constructs the Reeb graph in time $O(n \log n)$, where n is the number of edges in the triangulation used to represent the 2-manifold and the Morse function.

Keywords. Computational topology, visualization, 2-manifolds, Morse functions, level sets, Reeb graphs, loops, algorithms.

1 Introduction

We study Reeb graphs of Morse functions over 2-manifolds [13]. After motivating this topic and introducing the necessary definitions, we will state the results obtained in this paper.

Motivation. We are interested in the topology of smooth functions as a means to analyze and visualize intrinsic properties of geometric models and scientific data. Specifically, we study Reeb graphs, which expresses the connectivity of level sets. These graphs have been used in the past to construct data structures and user-interfaces for modeling and visualization applications. In computer-aided geometric design, Reeb graphs have been used to describe surface embeddings up to isotopy [18]. Applications of this idea include the evolution of teeth contact interfaces in the chewing process [17]. Multi-resolution versions of the Reeb graph have lead to data-base search methods for topologically similar geometric models [8]. In the interactive exploration of scientific data, Reeb graphs are used to efficiently compute level sets (iso-surfaces) [9]. Reeb graphs can also function as a user-interface tool aiding the selection of meaningful level sets [2]. A more extensive discussion of Reeb graphs and their variations in geometric modeling and visualization applications can be found in [6].

The first algorithm for constructing the Reeb graph of a smooth function over a 2-manifold is due to [16]. Given a triangulation of that 2-manifold, this algorithm takes as input all distinct level lines and thus may take time $O(n^2)$, where n is the number of vertices in the triangulation. An improvement of the running time at the cost of accuracy has been suggested in [8]. An $O(n \log n)$ time algorithm for loop-free Reeb graphs over manifolds of constant dimension has been described in [3]. For the case of 3-manifolds, this algorithm has been extended to include information about the genus of the level surfaces in [12]. In this paper, we focus on loops in Reeb graphs and study when they occur and how they can be constructed.

Definitions. Let $\mathbf{M}$ be a manifold with or without boundary. We are interested in smooth maps $f : \mathbf{M} \to \mathbb{R}$, and for simplicity in generic such maps known as *Morse functions*, which are defined by the following conditions:

I. all critical points of f are non-degenerate and lie in the interior of $\mathbf{M}$;

II. all critical points of f restricted to the boundary of $\mathbf{M}$ are non-degenerate; and

III. $f(x) \neq f(y)$ for all critical points $x \neq y$ of f and its restriction to the boundary.

Compare this with the definition of a Morse function of a stratified space in the book by Goresky and MacPhearson [7,

*Research of the second and forth authors is partially supported by NSF under grants EIA-99-72879, and CCR-00-86013. Research of the third author is partially supported by NSF under grant DMS-01-07621. This work was performed under the auspices of the U. S. Department of Energy by University of California Lawrence Livermore National Laboratory under contract No. W-7405-Eng-48.

†Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, Livermore, California.

‡Departments of Computer Science and Mathematics, Duke University, Durham, and Raindrop Geomagic, Research Triangle Park, North Carolina.

§Department of Mathematics, Duke University, Durham, North Carolina.

¶Department of Computer Science, Duke University, Durham, North Carolina.

Section I.4]. A *level set* is the preimage of a constant value, $f^{-1}(t)$. The *Reeb graph* of f is obtained by contracting the connected components of the level sets to points. An example for a 2-manifold without boundary is shown in Figure 1. Branching in the Reeb graph occurs only at nodes that cor-

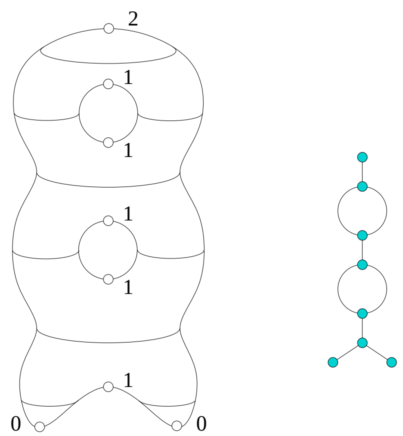

Figure 1: To the left: a double torus with f equal to the height or distance from a horizontal base plane. We have $k_0 = 2, k_1 = 5, k_2 = 1$ and therefore $\chi = -2$. To the right: the corresponding Reeb graph.

respond to level sets passing through critical points of f. We write k_i for the number of critical points of index i of f and $\chi = \sum_i (-1)^i k_i$ for the Euler characteristic of $\mathbf{M}$.

Results. Denote the number of loops in the Reeb graph R of f by $\#\text{loops} = \#\text{loops}_f$. This number is equal to the first Betti number, $\beta_1(R)$, which is the rank of the first homology group, $\mathsf{H}_1(R)$. We have a continuous surjection from $\mathbf{M}$ to R in which the loops in $\mathbf{M}$ that map to different loops in R are neither contractible nor equivalent. It follows that the first Betti number of $\mathbf{M}$ bounds the one of R from above:

$$\#\text{loops} \;=\; \beta_1(R) \;\leq\; \beta_1(\mathbf{M}). \qquad (1)$$

Depending on what $\mathbf{M}$ is, this bound may or may not be tight. We use combinatorial and topological arguments to get tight bounds for manifolds of dimension 2.

According to the classification theorem of 2-manifolds, see eg. [10, 15], a connected orientable 2-manifold $\mathbf{M}$ is either diffeomorphic to the sphere or the connected sum of $g \geq 1$ tori, and a non-orientable 2-manifold $\mathbf{N}$ is diffeomorphic to the connected sum of $g \geq 1$ projective planes. Denoting the sphere, the torus, and the projective plane by $\mathbf{S}$, $\mathbf{T}$, and $\mathbf{P}$, we write

$$\begin{aligned} \mathbf{M} &= \mathbf{S},\ \mathbf{T}\#\mathbf{T}\#\ldots\#\mathbf{T} \quad \text{and} \\ \mathbf{N} &= \mathbf{P}\#\mathbf{P}\#\ldots\#\mathbf{P}. \end{aligned}$$

The Klein bottle is $\mathbf{K} = \mathbf{P}\#\mathbf{P}$, and each connected non-orientable 2-manifold is also diffeomorphic to the connected sum of an orientable 2-manifold with $\mathbf{P}$ or $\mathbf{K}$. The *genus* is the maximum number of pairwise disjoint simple closed curves along which we can cut while keeping the manifold in one piece. For both families, the genus is equal to g. We have $g = (2-\chi)/2$ in the orientable case and $g = 2 - \chi$ in the non-orientable case. We get a 2-manifold with boundary by removing $h \geq 1$ disks from one without boundary. Table 1 summarizes our results on the number of loops in the Reeb graph.

mfld	g	$h=0$	$h=1$	$h=2$	$h=3$	$\ldots$
$\mathbf{S}$	0	0	0	$[0,1]$	$[0,2]$	$\ldots$
$\mathbf{T}$	1	1	$[1,2]$	$[1,3]$	$[1,4]$	$\ldots$
$\mathbf{T}\#\mathbf{T}$	2	2	$[2,4]$	$[2,5]$	$[2,6]$	$\ldots$
$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$
$\mathbf{P}$	1	0	$[0,1]$	$[0,2]$	$[0,3]$	$\ldots$
$\mathbf{K}$	2	$[0,1]$	$[0,2]$	$[0,3]$	$[0,4]$	$\ldots$
$\mathbf{K}\#\mathbf{P}$	3	$[0,1]$	$[0,3]$	$[0,4]$	$[0,5]$	$\ldots$
$\mathbf{K}\#\mathbf{K}$	4	$[0,2]$	$[0,4]$	$[0,5]$	$[0,6]$	$\ldots$
$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$

Table 1: Upper and lower bounds on the number of loops in the Reeb graphs of Morse functions over orientable and non-orientable 2-manifolds with and without boundary.

To study the computational complexity of constructing a Reeb graph, we assume a piecewise linear representation of the 2-manifold and the Morse function. Letting n be the number of edges of the triangulation, we give an algorithm that constructs the Reeb graph in time $O(n \log n)$.

Outline. Sections 2 and 3 prove the bounds on the number of loops for orientable and non-orientable 2-manifolds with and without boundary. Sections 4 and 5 present the algorithm for constructing Reeb graphs for piecewise linear input data. Section 6 concludes this paper.

2 Orientable 2-manifolds

In this section, we prove tight upper and lower bounds on the number of loops in the Reeb graphs of Morse functions over orientable 2-manifolds with and without boundary.

Without boundary. Let $\mathbf{M}$ be a connected 2-manifold without boundary and $f : \mathbf{M} \to \mathbb{R}$ a Morse function. A point of the Reeb graph R of f corresponds to a level set, and we refer to this point as a *node* if the level set passes through a critical point of f. The rest of the Reeb graph consists of arcs connecting the nodes. The *degree* of a node is the number of arcs that share the node. Since, by Condition III, all critical points have different function values, we have a bijection between the nodes of R and the critical points of f. A minimum starts and a maximum ends a family of contour cycles, which implies that they correspond to degree-1 nodes. A saddle either splits a single cycle into two or it merges two cycles into one, and in either case it corresponds to a degree-3 node. Higher degrees occur only for functions f that are not Morse.

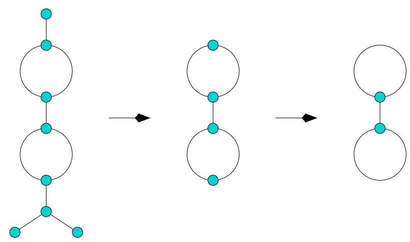

Figure 2: From left to right: the original Reeb graph of the double torus in Figure 1, the graph after collapsing, and the graph after merging arcs.

We collapse degree-1 nodes and merge arcs across degree-2 nodes. Both operations preserve the homotopy type and thus the number of loops in R. This is illustrated in Figure 2. Let R' be the Reeb graph after collapsing and merging. Let ℓ_3 be the number of degree-3 nodes in R' and note that ℓ_3 is even because $3\ell_3$ is twice the number of arcs.

`Case 1.` R' has at least one loop. We prove — either by induction or using the Euler formula for graphs — that the number of loops in R' is $\frac{\ell_3}{2} + 1$. We have k_1 degree-3 nodes in R, and for each minimum and maximum we collapse a degree-1 node turning a degree-3 into a degree-2 node, which is then removed. Hence, $\ell_3 = k_1 - (k_0 + k_2) = -\chi = 2g - 2$. Since ℓ_3 is a non-negative even integer, this covers the case in which $\mathbf{M}$ is the connected sum of $g \geq 1$ tori.

`Case 2.` R' has no loop and thus consists of a single node. Since all other cases are covered in the first case, $\mathbf{M}$ is the sphere.

By construction, $\#\mathrm{loops}_f$ is equal to the number of loops in R'. In both case, this number is equal to the genus:

LEMMA A. The Reeb graph of a Morse function over a connected orientable 2-manifold of genus g without boundaries has g loops.

A topologist's approach to proving Lemma A would be to observe that thickening the Reeb graph gives a 3-manifold $\mathbf{Q}$ whose boundary is the 2-manifold $\mathbf{M}$. Consider now the inclusion of the first homology groups, $\mathsf{H}_1(\mathbf{M}) \hookrightarrow \mathsf{H}_1(\mathbf{Q})$. By Lefschetz duality and Poincaré duality, the rank of the kernel is half the rank of the domain. We have $\mathrm{rank}\,\mathsf{H}_1(\mathbf{M}) = 2g$ and therefore $\mathrm{rank}\,\mathsf{H}_1(\mathbf{Q}) = 2g - g = g$, which is also the number of loops in the Reeb graph.

Note that both proofs make little use of the properties of Morse functions, implying that Lemma A holds for more general but not for arbitrary continuous functions.

With boundary. Let $\mathbf{M}$ be a connected orientable 2-manifold of genus g with $h \geq 1$ boundary components. The Euler characteristic is $\chi = 2 - 2g - h$, which implies $\beta_1 = 2g + h - 1$. The inequality (1) implies that this is also an upper bound on the number of loops. We complement this with a lower bound to get:

LEMMA B. The Reeb graph of a Morse function over a connected orientable 2-manifold of genus g with $h \geq 1$ boundary components has between g and $2g + h - 1$ loops.

PROOF. Let f be a Morse function over $\mathbf{M}$. To prove the lower bound, we show that it is possible to close one hole by removing one boundary component without increasing the number of loops in the corresponding Reeb graph. Every boundary component is a circle and the restriction of f to that circle alternates between local minima and maxima. Let q be a maximum with neighboring minima p and r. If $p = r$, we glue the two arcs by identifying points with equal function values. Otherwise, assume $f(p) < f(r)$ and let r' be the point along the arc qp with $f(r') = f(r)$. Again we glue the arcs qr' and qr by identifying points with equal function values. In both cases, the gluing starts from a single point and thus does not create any new loops (loops can only be removed). Indeed, the effect of the operation on the Reeb graph is either void or that of a zipper merging portions of two arcs starting at a common endpoint. By repeating the gluing operation we eventually remove the hole. After eliminating all h holes, we are left with a function over a 2-manifold without boundary. The genus is still g, so by Lemma A, the Reeb graph has g loops. By construction, the Reeb graph of f has at least that number of loops. ◻

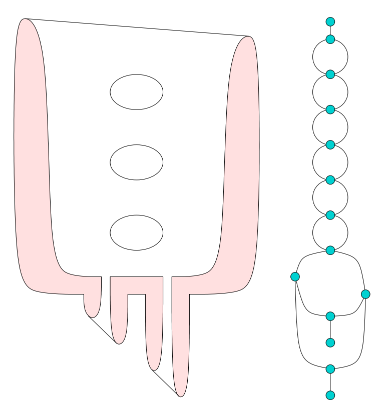

Figure 3: We get two loops per tunnel and one loop per strip.

The bounds in Lemma B are tight. To realize the lower bound of g loops, we start with a 2-manifold without boundary and cut h holes in sequence, such that no two boundary components meet a common contour cycle. To realize the upper bound of $2g - h + 1$ loops we follow the recipe illustrated in Figure 3. We start with a disk that hangs like a blanket over the back of a chair. We add g tunnels in sequence and finally add $h - 1$ strips at the bottom to increase the number of boundary components to h.

3 Non-orientable 2-manifolds

In this section, we prove tight upper and lower bounds on the number of loops in the Reeb graphs of Morse functions over non-orientable 2-manifolds with and without boundary.

Without boundary. Each connected non-orientable 2-manifold without boundary is the connected sum of g copies of the projective plane: $\mathbf{N} = \mathbf{P}\#\mathbf{P}\#\ldots\#\mathbf{P}$. Recall that $\mathbf{K} = \mathbf{P}\#\mathbf{P}$ is the Klein bottle and that g is the genus of $\mathbf{N}$. The Euler characteristic of $\mathbf{N}$ is $\chi = 2-g$. We make use of a particular 2-sheeted cover of $\mathbf{N}$ obtained by doubling every point $x \in \mathbf{N}$ into a pair x' and x'', which we imagine as lying near x and locally on opposite sides of the non-orientable manifold. Two points x' and y' are *near* if $x, y \in \mathbf{N}$ are near and x', y' lie locally on the same side of $\mathbf{N}$. The resulting space is a connected and orientable 2-manifold $\mathbf{M}$ with Euler characteristic $\chi_{\mathbf{M}} = 2\chi$. The genus of $\mathbf{M}$ is therefore $g_{\mathbf{M}} = (2-\chi_{\mathbf{M}})/2 = 1-\chi = g-1$. Table 2 illustrates the correspondence between the non-orientable and the orientable 2-manifolds.

χ	g	$\mathbf{N}$	$\mathbf{M}$	$g_{\mathbf{M}}$	$\chi_{\mathbf{M}}$
1	1	$\mathbf{P}$	$\mathbf{S}$	0	2
0	2	$\mathbf{K}$	$\mathbf{T}$	1	0
-1	3	$\mathbf{K}\#\mathbf{P}$	$\mathbf{T}\#\mathbf{T}$	2	-2
…	…	…	…	…	…

Table 2: Doubling turns the non-orientable 2-manifold on the left into the orientable 2-manifold in the same row on the right.

Let R be the Reeb graph of $\mathbf{N}$. We have seen that R has degree-1 nodes corresponding to minima and maxima and degree-3 nodes corresponding to saddles of the Morse function over $\mathbf{N}$. In the non-orientable case, we also have degree-2 nodes, that correspond to what might be called *confused saddles*, as illustrated in Figure 4. To determine the effect of

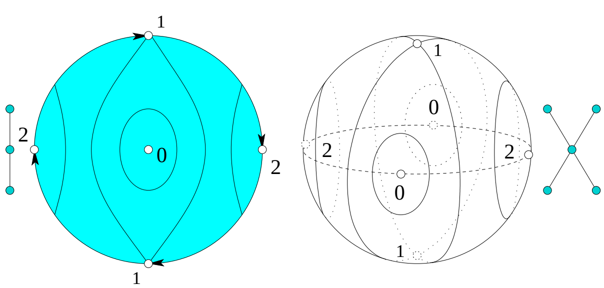

Figure 4: The height function over the projective plane on the left has a confused saddle, which corresponds to a degree-2 node in the Reeb graph. The 2-sheeted cover on the right is a sphere with degenerate height function.

doubling on contour cycles, we follow the two copies of each cycle around $\mathbf{N}$ to see how they are connected. This way we find that doubling turns every contour cycle into two, except if it passes through a confused saddle, in which case doubling generates four branches glued at a double-point, which is the image of the confused saddle. The double-point can be separated into two different points by turning the clover-like pattern into one of a highway interchange, in which the middle intersection is resolved into an overpass crossing. The Reeb graph R' of $\mathbf{M}$ is obtained by connecting two copies of R. One way to connect is by gluing the two copies at corresponding degree-2 nodes, another is by re-connecting arcs at degree-3 nodes. The latter happens for example when we double the Klein bottle, as in Figure 5. Formally, a *re-connection* exchanges the corresponding end-

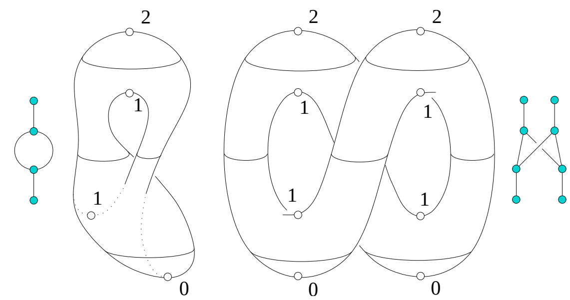

Figure 5: The Klein bottle with Reeb graph R on the left and the torus obtained by doubling with its Reeb graph R' on the right.

points of two corresponding arcs while keeping the other two endpoints the same. We thus obtain R' from two copies of R by a non-negative number of re-connections and $\ell_2 \geq 0$ gluing operations. If $\ell_2 = 0$ then the first re-connection connects the two copies and thus removes one loop. Every additional re-connection keeps the number of loops the same. If $\ell_2 > 0$ then the first gluing operation connects the two copies of R and every additional one adds a loop. Additional re-connections keep the number of loops the same. In either case, we have $\#\text{loops}_f = 2\#\text{loops}_e + (\ell_2 - 1)$, where e is the Morse function over $\mathbf{N}$ and f is the doubled Morse function over $\mathbf{M}$. Hence,

$$\begin{aligned} \#\text{loops}_e &= (\#\text{loops}_f - \ell_2 + 1)/2 \\ &= (g_{\mathbf{M}} + 1 - \ell_2)/2 \\ &= (g - \ell_2)/2. \end{aligned}$$

Since the number of loops is a non-negative integer, we have $\ell_2 \leq g$ and $\ell_2 \equiv g \pmod 2$. Table 3 shows the sets of possibilities for the first five non-orientable 2-manifolds.

$\mathbf{N}$	g	ℓ_2	#loops
$\mathbf{P}$	1	1	0
$\mathbf{K}$	2	$0, 2$	$1, 0$
$\mathbf{K}\#\mathbf{P}$	3	$1, 3$	$1, 0$
$\mathbf{K}\#\mathbf{K}$	4	$0, 2, 4$	$2, 1, 0$
$\mathbf{K}\#\mathbf{K}\#\mathbf{P}$	5	$1, 3, 5$	$2, 1, 0$
…	…	…	…

Table 3: The possible numbers of confused saddles and loops for non-orientable 2-manifolds.

LEMMA C. The Reeb graph of a Morse function over a connected non-orientable 2-manifold of genus g without boundary has between 0 and $\lfloor \frac{g}{2} \rfloor$ loops.

The bounds for the number of loops are tight. Note that for the projective plane, we have $\ell_2 = 1$ and $\#\mathrm{loops} = 0$, independent of the Morse function. For general $\mathbf{N}$, we get $\#\mathrm{loops} = 0$ by connecting g copies of $\mathbf{P}$ in a chain, as shown in Figure 6 but without boundary cycles. We get $\#\mathrm{loops} = \lfloor \frac{g}{2} \rfloor$ by connecting $\lfloor \frac{g}{2} \rfloor$ copies of the Klein bottle as in Figure 5, possibly by adding another copy of $\mathbf{P}$. Numbers between the two extremes are obtained by combining the two constructions.

With boundary. Same as in the orientable case, a connected non-orientable 2-manifold $\mathbf{N}$ with boundary is characterized up to diffeomorphism by its genus g and the number $h \geq 1$ of boundary components. Its Euler characteristic is $\chi = 2 - g - h$. The inequality (1) implies that $\beta_1 = 1 - \chi = g + h - 1$ is an upper bound on the number of loops in the Reeb graph:

LEMMA D. The Reeb graph of a Morse function over a connected non-orientable 2-manifold of genus g with $h \geq 1$ boundary components has between 0 and $g + h - 1$ loops.

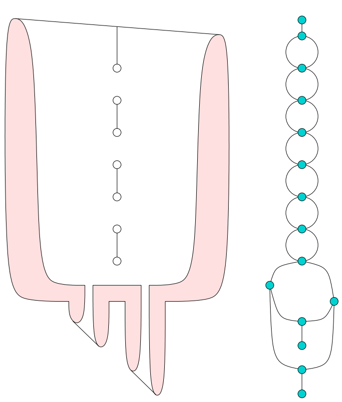

Figure 6: We get one loop per cross-cap and one per strip. The confused saddles delimiting the cross-caps are shown as small circles.

The bounds in Lemma D are tight. To see the lower bound take a Morse function over a non-orientable 2-manifold without boundary whose Reeb graph has no loop and cut h holes in sequence, such that no two boundary components meet a common contour cycle. To show that the upper bound is tight, we use the construction of Figure 3 and replace tunnels by double cross-caps and, if g is odd, add a single cross-cap at the top. As illustrated in Figure 6, this is done so all confused saddles have smaller function values than the blanket at the left and right shoulders.

4 Sweep Algorithm

In this section, we describe an algorithm that constructs the Reeb graph of a function $f : \mathbf{M} \to \mathbb{R}$. We remain on an abstract level and defer implementation details to Section 5.

Scheduling critical points. We construct the Reeb graph by maintaining the level set while sweeping $\mathbb{R}$ from $-\infty$ to $+\infty$. By definition of Morse functions, the critical points have pairwise different function values. It follows we can order them as $p_1, p_2, \ldots, p_k$ such that $f(p_i) < f(p_j)$ whenever $i < j$. Define $f(p_0) = -\infty$ and $f(p_{k+1}) = \infty$ and choose a value t_i strictly between the function values of p_i and p_{i+1}, for each i. Let $L_i = f^{-1}(t_i)$ be the corresponding level sets. In general, L_i consists of finitely many closed curves (contour cycles) and curves with endpoints (contour paths). The level set remains topologically the same as long as the function value remains strictly between $f(p_i)$ and $f(p_{i+1})$. The level sets in the next open interval are represented by L_{i+1}. The difference between L_i and L_{i+1} can be understood by studying how the level set changes when it passes through p_{i+1}. We distinguish between critical points of f and critical points of f restricted to the boundary of $\mathbf{M}$.

If p_{i+1} is a minimum of f then L_{i+1} develops a new contour cycle. Without that cycle, L_{i+1} would be topologically the same as L_i. If p_{i+1} is a maximum of f then L_{i+1} loses one contour cycle. Without that cycle, L_i would be topologically the same as L_{i+1}. If p_{i+1} is a saddle, the evolution from L_i to L_{i+1} is more complicated because it depends on the global and not just the local connectivity of the level set. Figure 7 illustrates the various cases that can occur. On the

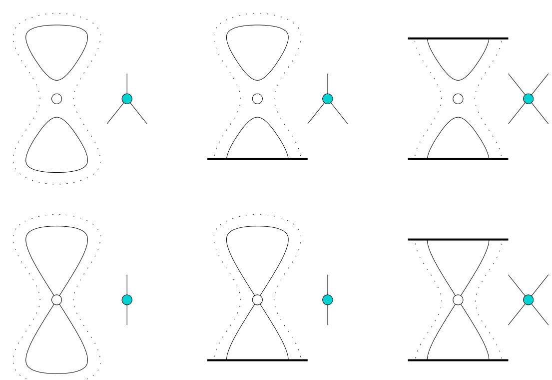

Figure 7: Evolution of the level set while passing through a (normal) saddle in the top row and a confused saddle in the bottom row. In each case, we show the corresponding new piece of the Reeb graph on the side. The solid and dotted lines show the level set before and after passing the saddle, and we get symmetric pictures (and upside down Reeb graph pieces) by switching solid with dotted lines.

boundary, we have two cases per minimum and maximum. If p_{i+1} is a boundary minimum, then L_{i+1} either develops a new contour path or it splits a contour cycle or path open, as illustrated in Figure 8. We have the symmetric cases for boundary maxima.

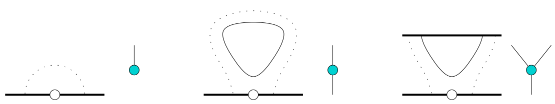

Figure 8: Evolution of the level set while passing through a boundary critical point. If that point is a minimum, the level set progresses from solid to dotted lines and the Reeb graph from bottom to top. For a boundary maximum the progress is the other way round.

Abstract data type. We store a level set as a collection of contour cycles and paths. The sweep is translated into a sequence of operations manipulating this collection. These operations create, destroy, cut, and glue cycles and paths:

MKCYCLE and MKPATH create new contour cycles and contour paths.

RMCYCLE and RMPATH destroy old contour cycles and contour paths.

CUT splits a cycle open to a path or a path into two paths.

GLUE connects two ends of a path to form a cycle or two ends of two different paths to form one path.

To determine the effect of a cut operation, we need to be able to tell a cycle from a path. Similarly, to determine the effect of a glue operation, we need to be able to tell whether two endpoints belong to the same or to two different paths. Assuming that ability, we have an algorithm that constructs the Reeb graph of $f : \mathbf{M} \to \mathbb{R}$ by maintaining the level set during a sweep. Each interior minimum/maximum translates into a cycle creation/destruction, each interior saddle translates into two cut and two glue operations, and each boundary minimum/maximum translates into a creation/destruction or a cut/glue operation. In total, we have at most $4k$ operations.

5 PL Implementation

In this section, we make the sweep algorithm of Section 4 concrete by assuming a piecewise linear input representation. It applies to the case common in practice, in which a function is only probed at a finite set of locations.

PL representation. A *triangulation* of a manifold $\mathbf{M}$ is a simplicial complex K whose underlying space is homeomorphic to $\mathbf{M}$; see eg. [1, 15]. Writing $\mathrm{Vert}\, K$ for the collection of vertices, we use a map $\varphi : \mathrm{Vert}\, K \to \mathbb{R}$ and define the function f over the 2-manifold $\mathbf{M}$ by linear extension over the simplices. It is clear that f is not smooth and thus definitely not a Morse function. We can, however, modify K and φ such that f has combinatorial properties that resemble the generic smooth properties of Morse functions. We need some definitions to explain this. The *star* of a vertex u consists of all simplices that share u, the *link* consists of all faces of simplices in the star that do not belong to the star themselves, and the *lower link* is the subset of the link induced by vertices with function value less than u:

$$\begin{aligned} \mathrm{St}\, u &= \{\sigma \in K \mid u \in \sigma\}, \\ \mathrm{Lk}\, u &= \{\tau \in K \mid \tau \subseteq \sigma \in \mathrm{St}\, u, \tau \notin \mathrm{St}\, u\}, \\ \underline{\mathrm{Lk}}\, u &= \{\tau \in \mathrm{Lk}\, u \mid v \in \tau \Longrightarrow f(v) < f(u)\}. \end{aligned}$$

We use the link and the lower link to classify the vertices into the various types of critical points we observe in the smooth setting. Consider first an interior vertex u. Its link is the cycle of edges and vertices surrounding u. We call u *regular* if the lower link is a non-empty connected segment of the link; in all other cases, u is *critical*. Specifically, u is a *minimum* if $\underline{\mathrm{Lk}}\, u = \emptyset$, u is a *maximum* of $\underline{\mathrm{Lk}}\, u = \mathrm{Lk}\, u$, and u is a *k-fold saddle* if $\underline{\mathrm{Lk}}\, u$ consists of $k+1 \geq 2$ components along the link. This is illustrated in Figure 9. Consider second a boundary vertex u. Its link is a half-circle. Along the boundary, u has two neighbors v and w, which are the endpoints of the half-circle. Assuming $f(v) < f(w)$, we call u a *boundary*

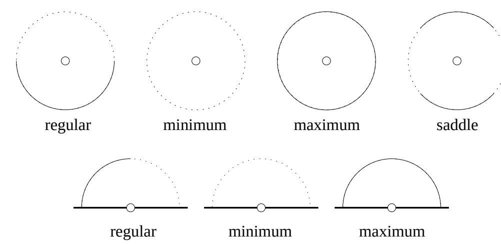

Figure 9: Regular and critical points as classified by the (solid) lower link, which is a subset of the (otherwise dotted) link.

$$\left.\begin{array}{l} \textit{regular point} \\ \textit{minimum} \\ \textit{maximum} \end{array}\right\} \quad \text{if} \quad \left\{\begin{array}{l} f(v) < f(u) < f(w), \\ f(v) > f(u) < f(w), \\ f(v) < f(u) > f(w). \end{array}\right.$$

Figure 9 illustrates only the simple cases. Non-simple cases reduce to simple ones. For example, a k-fold saddle can be split into k simple saddles, as described in [5]. Similarly, a boundary minimum/maximum with more complicated lower link than shown in Figure 9 can be split into simple (interior) saddles and a (simple) boundary minimum/maximum. Note that we just eliminated the rightmost two cases in Figure 8. The corresponding smooth operation bends $\mathbf{M}$ a little near the boundary and thus creates a saddle right next to the boundary minimum/maximum. In addition to having all non-simple critical points split into simple ones, we assume $\varphi(u) \neq \varphi(v)$ for all vertices $u \neq v$ of K.

PL algorithm. The combinatorial structure of the piecewise linear representation of f is now simple enough to allow a direct implementation of the sweep algorithm. The sweep translates into processing the vertices of K in the sequence of increasing function values. A generic level set consists of finitely many contour cycles and contour paths. We store each cycle as a cyclic list and each path as a linear list of edges in K. These edges carry the vertices of the contours, and the triangles between them carry their edges. We now discuss how the collection of cyclic and linear lists changes as we pass through a vertex u of K. If u is regular, we just replace its descending edges (which all belong to a single list) by its ascending edges. In addition to the operations listed in Section 4, we therefore need the following two:

DELETE removes an edge from a cyclic or linear list.

INSERT adds an edge to a cyclic or linear list.

Similarly, if u is critical, we remove its descending edges and add its ascending ones. However, now the level set changes its topology, which is reflected by structural changes in the collection of lists caused by the cut and glue operations, as described in Section 4. To identify structural changes, we use yet another operation:

FIND determines the list that contains a given edge of K.

The effect of the cut and glue operations is determined by finding the lists from which the descending edges are removed and to which the ascending edges are added. By restriction to simple critical vertices, these edges belong to either one or to two lists. The corresponding local pieces appended to the Reeb graph are shown in Figure 10.

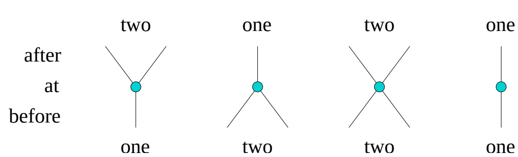

Figure 10: Adding nodes of degree 3, 3, 4, and 2 to the Reeb graph; compare with Figure 7.

Data structure and analysis. To implement all operations efficiently, we represent each list (cyclic and linear) as a balanced search tree [4]. Each insertion and deletion takes time at most logarithmic in the size of the list. Finding a given edge is implemented by walking upward to the root of the tree, which also takes at most logarithmic time. Finally, cut and glue operations are implemented by splitting and concatenating trees. Here it is important that each list is marked as either cyclic or linear, because cutting a linear list translates into splitting to generate two linear lists, while cutting a cyclic list translates into splitting and concatenating the two pieces to generate a single linear list. Similar distinctions have to be made when we glue two ends of one or two linear lists. Standard balanced search trees support split and concatenate operations again in logarithmic time. The randomized search trees developed by Seidel and Aragon [14] are particularly well suited for our purposes because they lend themselves to rather simple implementations of all necessary operations. Furthermore, experiments indicate that they outperform all alternative data structures in practice.

To give a bound on the running time of the above algorithm, we let n be the number of edges in K. Note that K contains fewer than n triangles and fewer than n vertices. We use n insertions, n deletions, and $2n$ find operations to process all edges. Furthermore, we use at most two cut and two glue operations per vertex. Each operation takes time at most $O(\log n)$, which adds to a total time of at most $O(n \log n)$ to construct the Reeb graph of f. If we use randomized search trees we get the same running time but only in the expected and not the worst case.

6 Discussion

In this paper, we study Reeb graphs of Morse functions over 2-manifolds with and without boundary. The number of loops is these graphs depends on the topology of the 2-manifold and sometimes but not always on the Morse function. We prove tight upper and lower bounds on the number of loops and give an $O(n \log n)$ time algorithm to construct a Reeb graph. It would be interesting to prove that $O(n \log n)$ is indeed optimal or, alternatively, to improve the algorithm. Maybe one of the lower bounds in [11] can be extended to prove the optimality of our algorithm.

Our understanding of Reeb graphs for 3-manifolds is much less developed. It is no longer true that the number of loops for an orientable manifold without boundary is independent of the function. For example, a 3-torus obtained by gluing opposite faces of a cube can have zero or one loop in the Reeb graph. We also have no algorithm for constructing the Reeb graph of a Morse function over a 3-manifold that runs in time less than quadratic in the size of the representing triangulation.

References

[1] P. S. ALEXANDROV. *Combinatorial Topology.* Republication of three volumes published in 1956, 57, 60, Dover, Mineola, New York, 1998.

[2] C. L. BAJAJ, V. PASCUCCI AND D. R. SCHIKORE. The contour spectrum. *In* "Proc. IEEE Visualization, 1997", 167–175.

[3] H. CARR, J. SNOEYINK AND U. AXEN. Computing contour trees in all dimensions. *Comput. Geom. Theory Appl.* **24** (2002), 75–94.

[4] T. H. CORMEN, C. E. LEISERSON AND R. L. RIVEST. *Introduction to Algorithms.* MIT Press, Cambridge, Massachusetts, 1994.

[5] H. EDELSBRUNNER, J. HARER AND A. ZOMORODIAN. Hierarchical Morse complexes for piecewise linear 2-manifolds. *Discrete Comput. Geom.*, to appear.

[6] A. T. FOMENKO AND T. L. KUNII, (EDS). *Topological Methods for Visualization.* Springer-Verlag, Tokyo, Japan, 1997.

[7] M. GORESKY AND R. MACPHEARSON. *Stratified Morse Theory.* Springer-Verlag, Heidelberg, Germany, 1988.

[8] M. HILAGA, Y. SHINAGAWA, T. KOMURA AND T. L. KUNII. Topology matching for fully automatic similarity estimation of 3D shapes. *In* "Computer Graphics Proc., SIGGRAPH 2001", 203–212.

[9] M. VAN KREVELD, R. VAN OOSTRUM, C. L. BAJAJ, V. PASCUCCI AND D. R. SCHIKORE. Contour trees and small seed sets for isosurface traversal. *In* "Proc. 13th Ann. Sympos. Comput. Geom., 1997", 212–220.

[10] W. S. MASSEY. *Algebraic Topology: An Introduction.* Springer-Verlag, New York, 1967.

[11] K. MEHLHORN, S. NÄHER AND H. ALT. A lower bound on the complexity of the union-split-find problem. *SIAM J. Comput.* **17** (1988), 1093–1102.

[12] V. PASCUCCI AND K. COLE-MCLAUGHLIN. Efficient computation of the topology of level sets. *Algorithmica*, to appear.

[13] G. REEB. Sur les points singuliers d'une forme de Pfaff complèment intégrable ou d'une fonction numérique. *Comptes Rendus de L'Académie ses Séances, Paris* **222** (1946), 847–849.

[14] R. SEIDEL AND C. ARAGON. Randomized search trees. *Algorithmica* **16** (1996), 464–497.

[15] H. SEIFERT AND W. THRELFALL. *Lehrbuch der Topologie.* Chelsea, New York, 1934.

[16] Y. SHINAGAWA AND T. L. KUNII. Constructing a Reeb graph automatically from cross sections. *IEEE Comput. Graphics Appl.* **11** (1991), 44–51.

[17] Y. SHINAGAWA, T. L. KUNII, H. SATO AND M. IBUSUKI. Modeling contact of two complex objects: with an application to characterizing dental articulations. *Computers and Graphics* **19** (1995), 21–28.

[18] S. YOSHIHISA, T. L. KUNII AND Y. L. KERGOSIAN. Surface coding based on Morse theory. *IEEE Comput. Graphics Appl.* **11** (1991), 66–78.